

FIG. 1

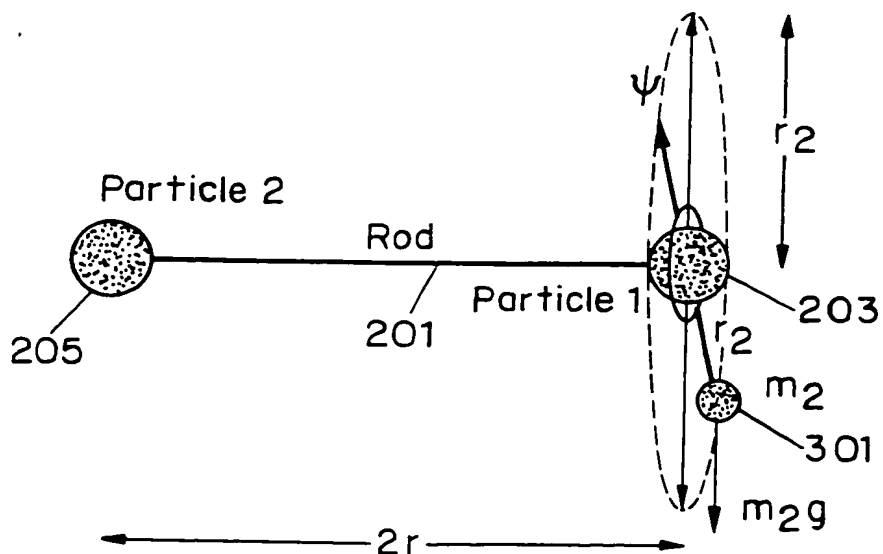


FIG. 3

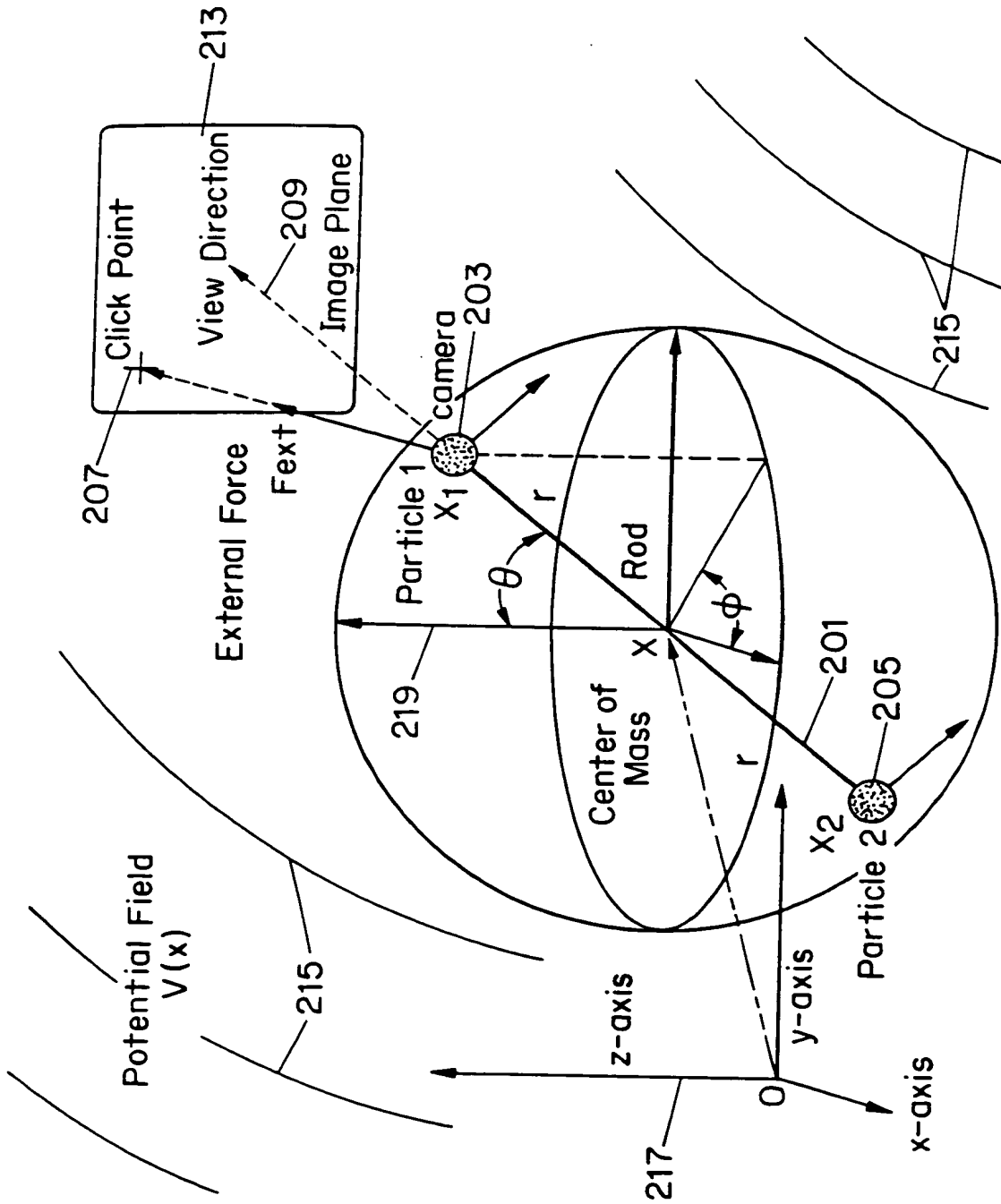


FIG. 2

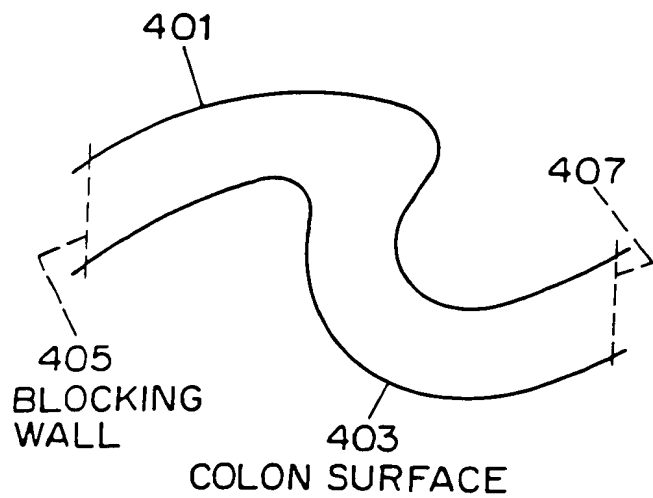


FIG. 4

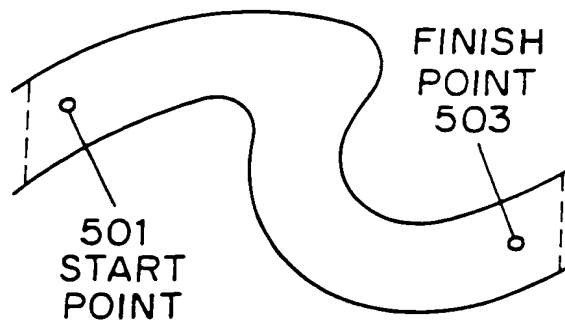


FIG. 5

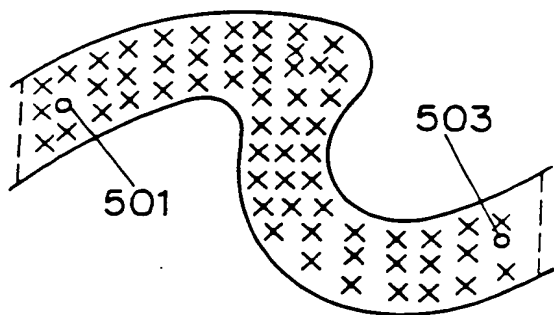


FIG. 6

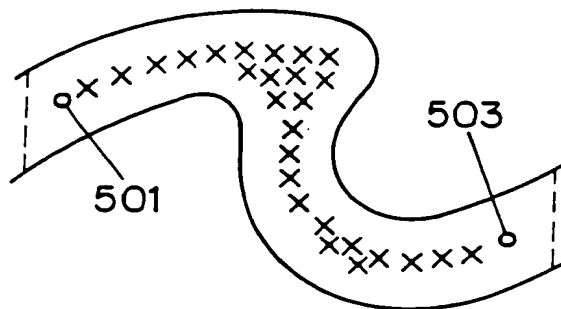


FIG. 7

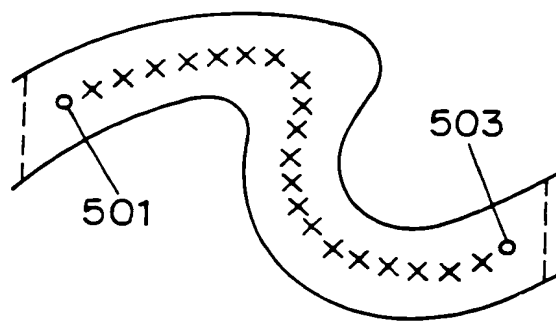


FIG. 8

0977460-0001  
T06020-02T2260

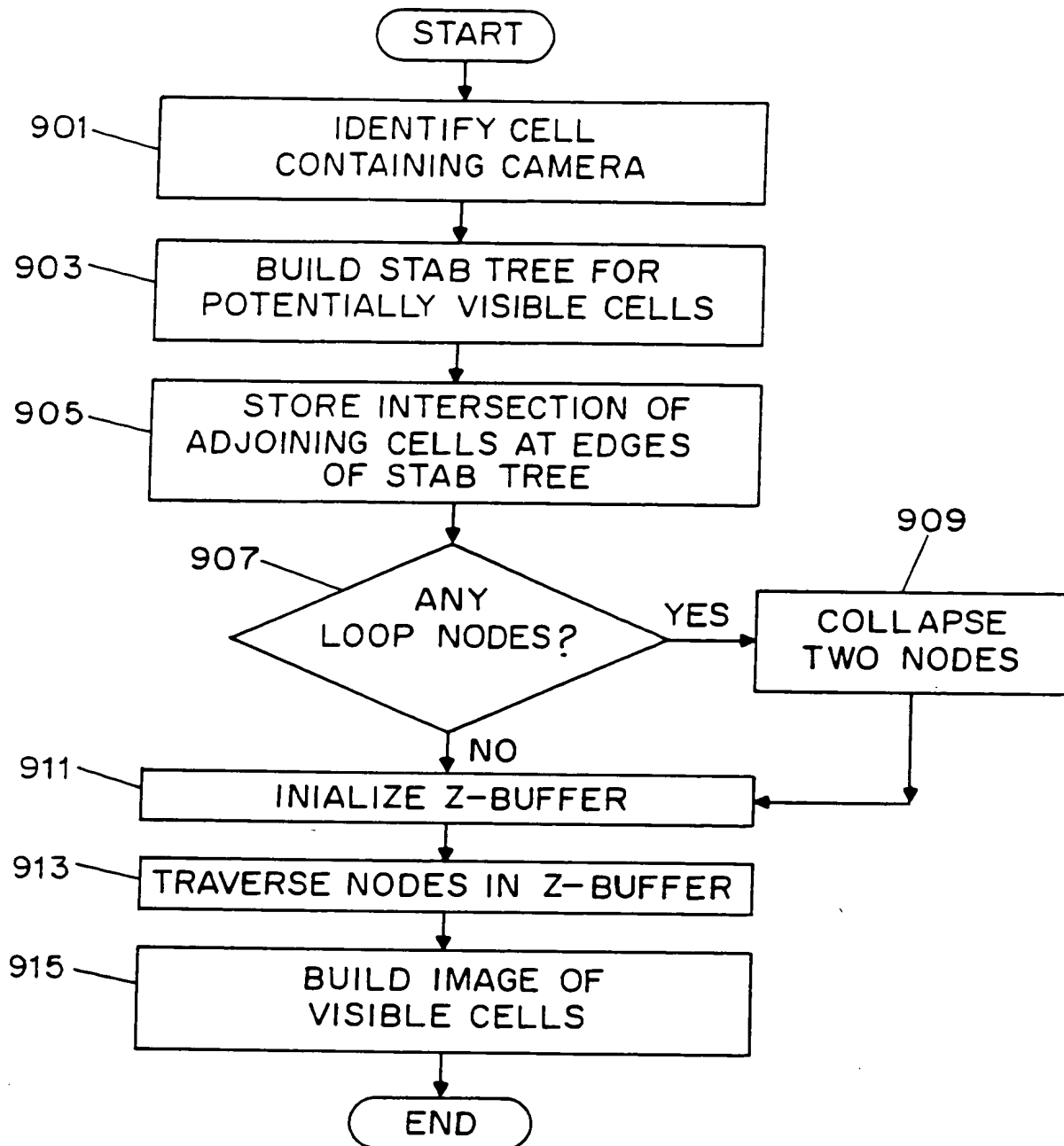


FIG. 9

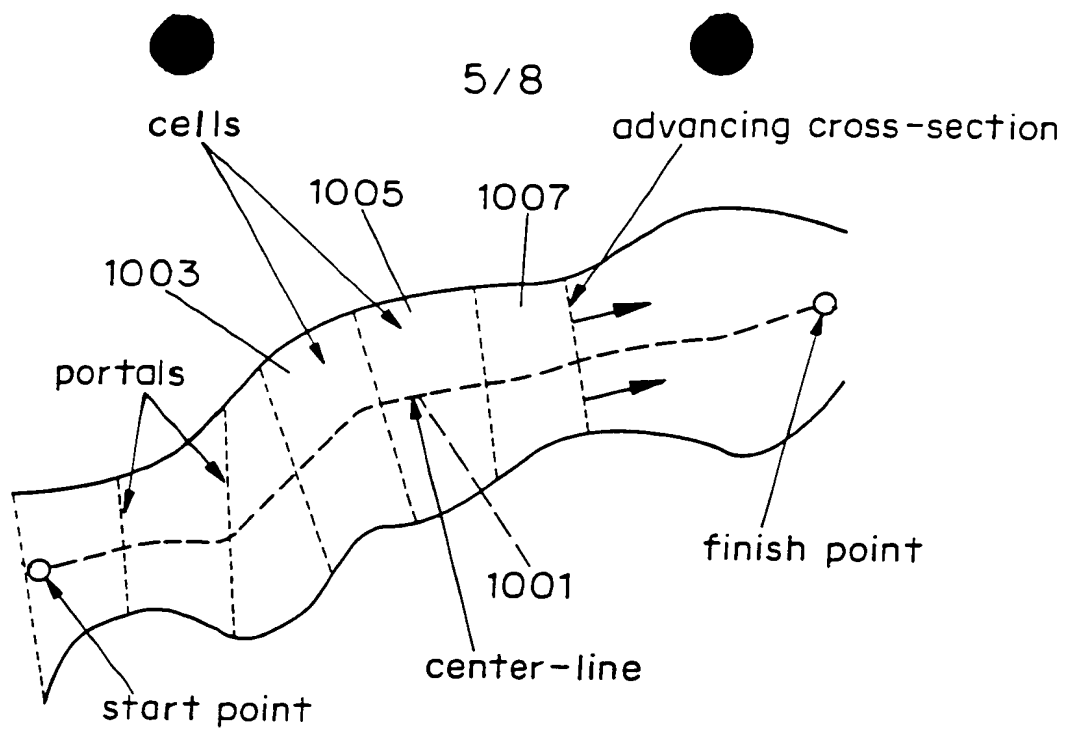


FIG. 10

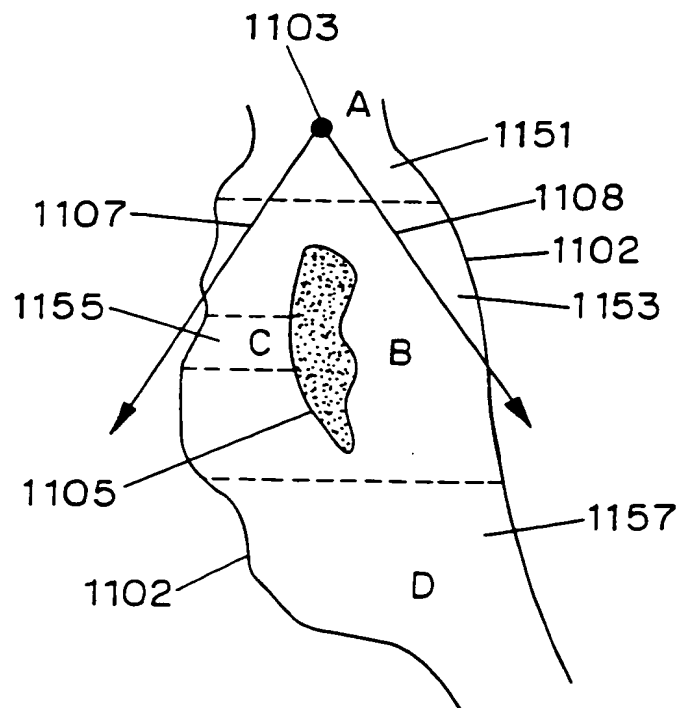


FIG. 11(a)

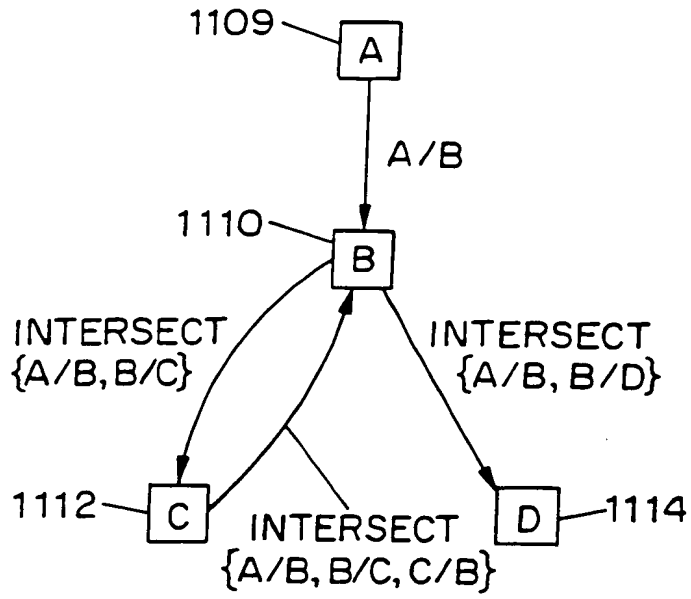


FIG. 11(b)

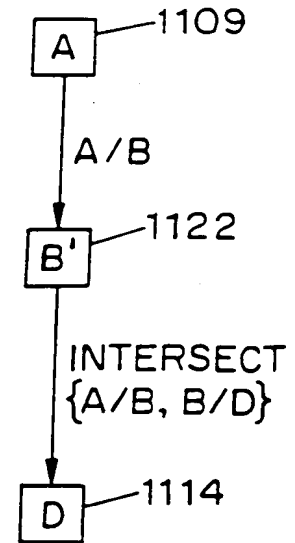


FIG. 11(c)

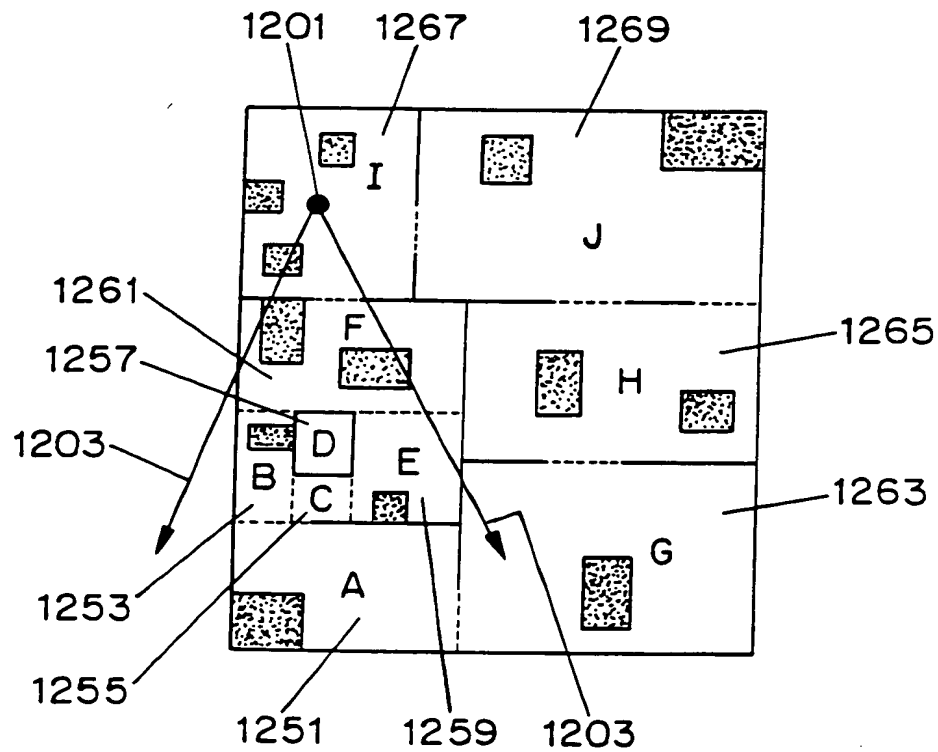


FIG. 12(a)

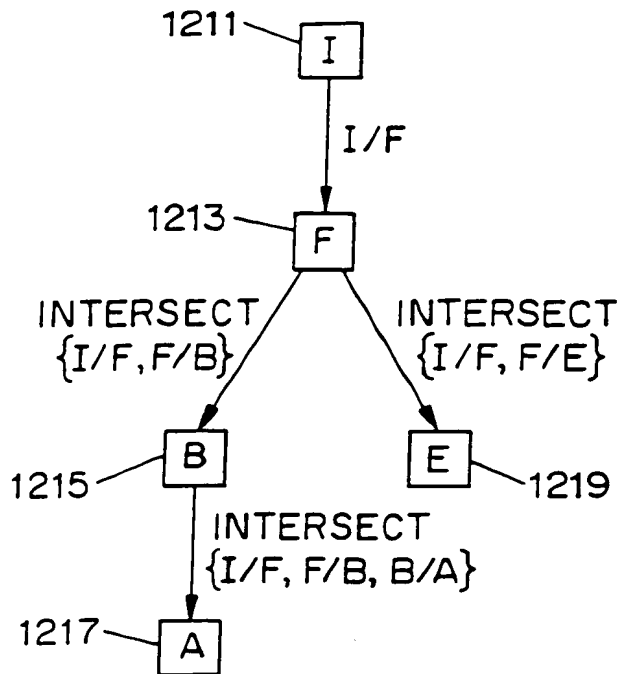


FIG. 12(b)

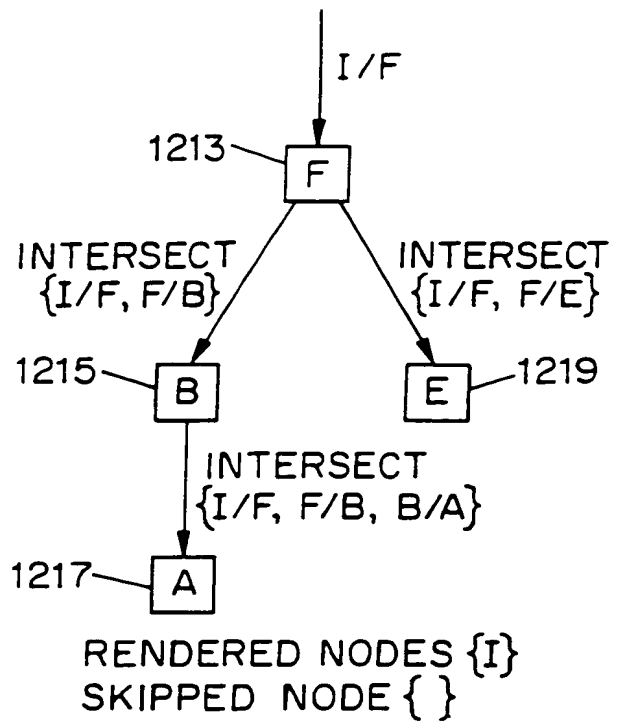


FIG. 12(c)

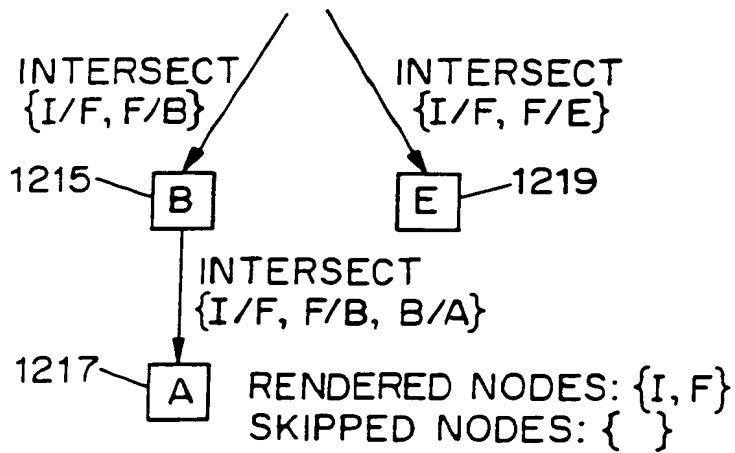


FIG. 12(d)

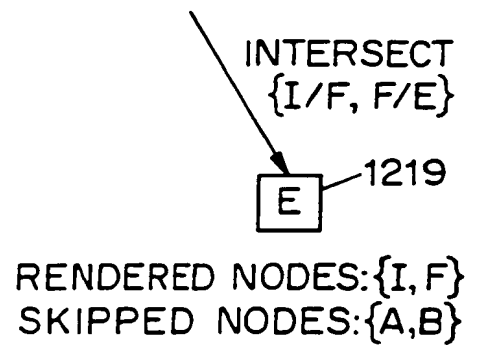


FIG. 12(e)

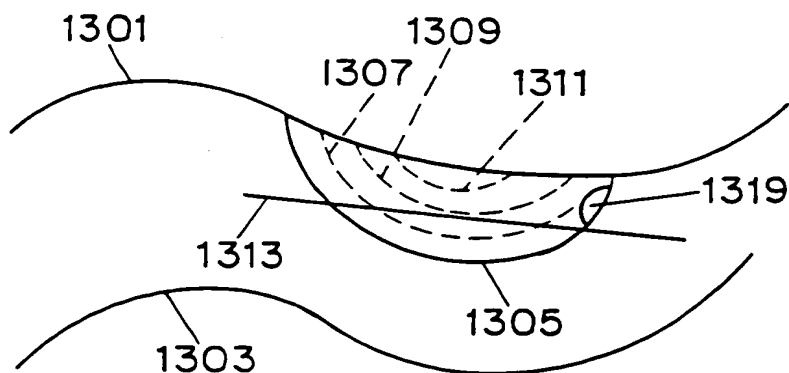


FIG. 13

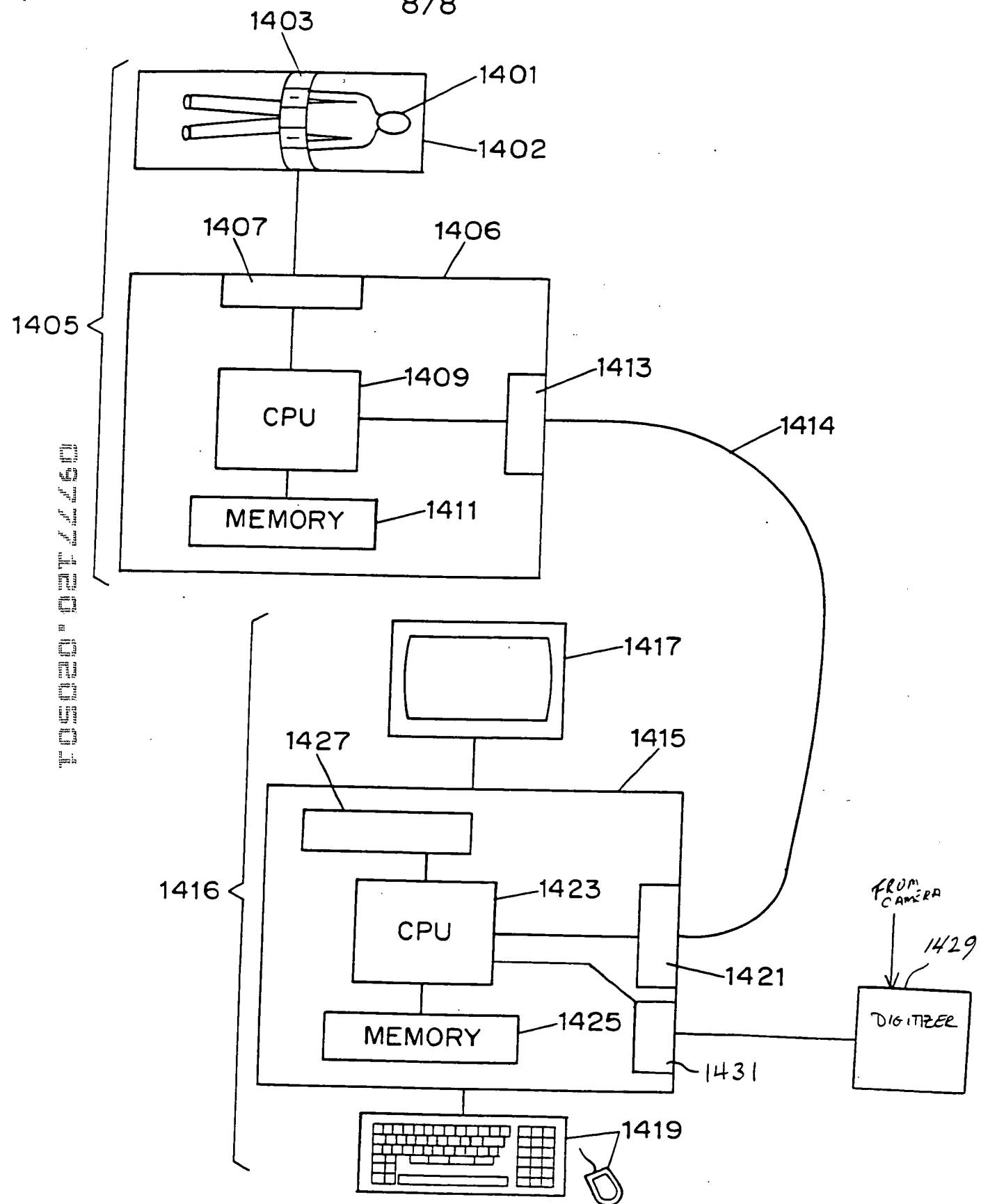
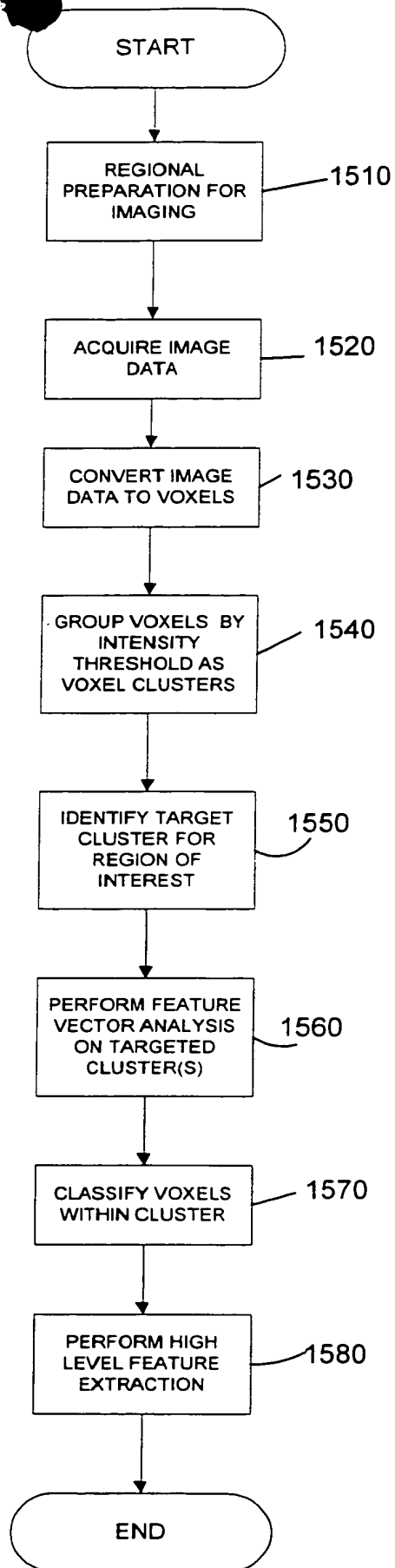


FIG. 14



Fig. 15



20250307 09:44:59

FIG. 16

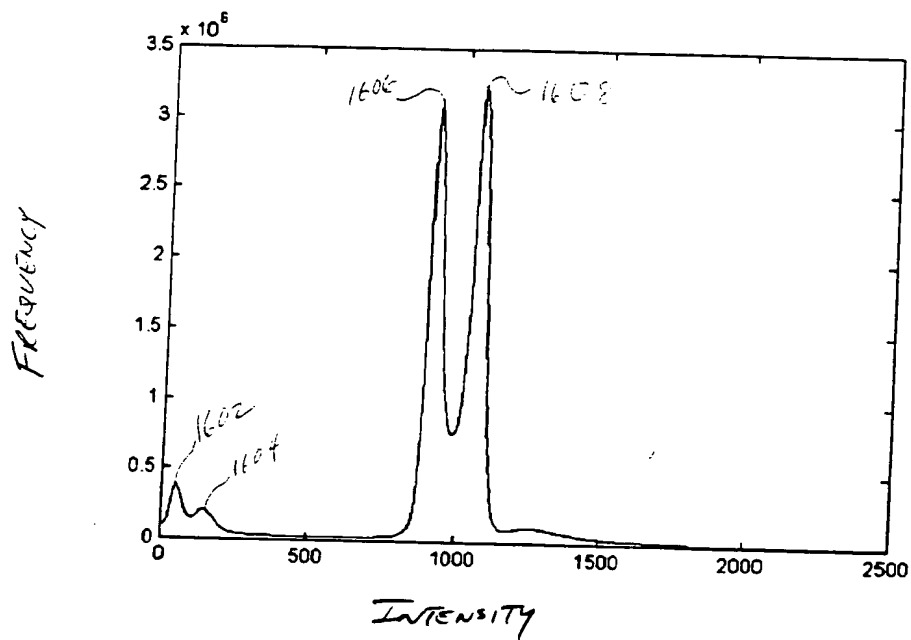
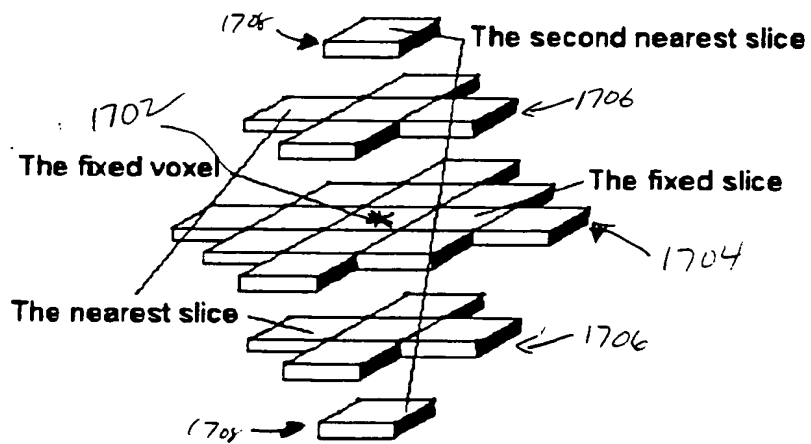


FIG. 17



F. 180

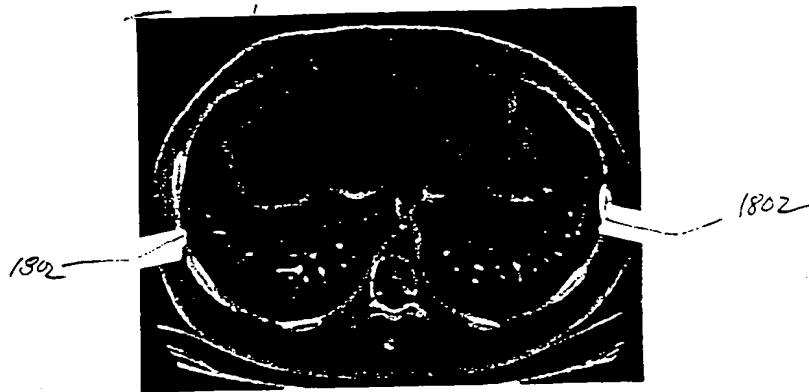


Fig 18B

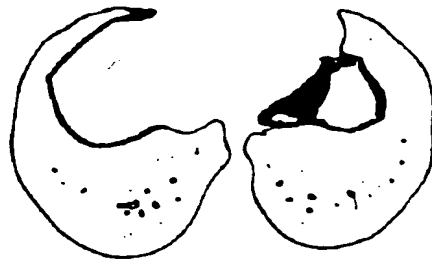


FIG. 18C

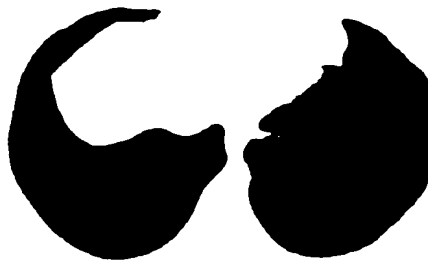


Fig 19A

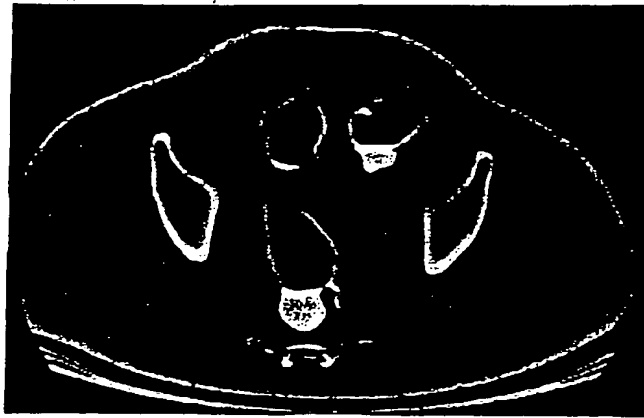


FIG. 19B

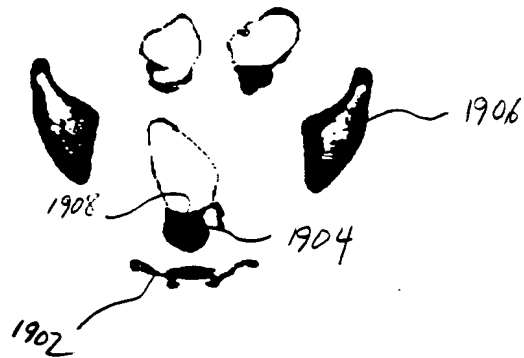


FIG 19C



2025020" 0247/260

Fig. 20

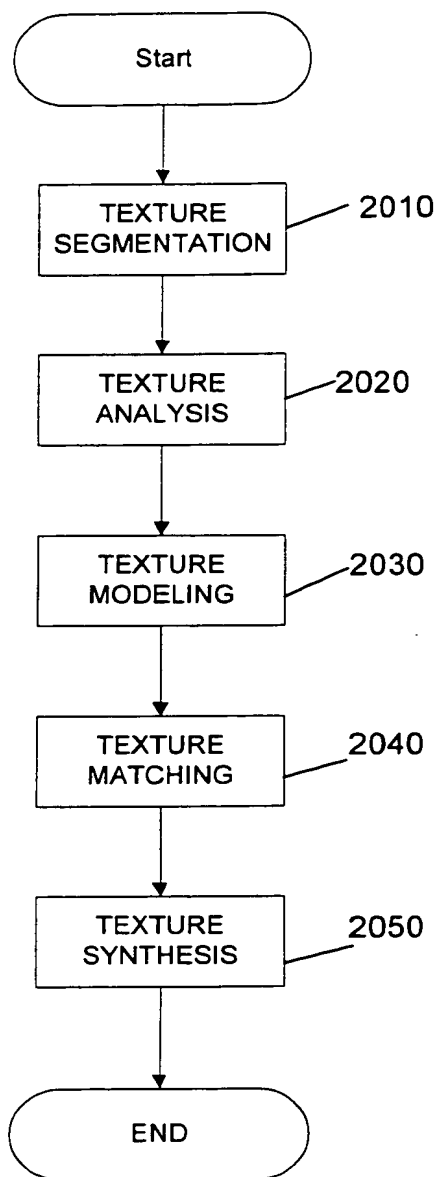


FIG. 21

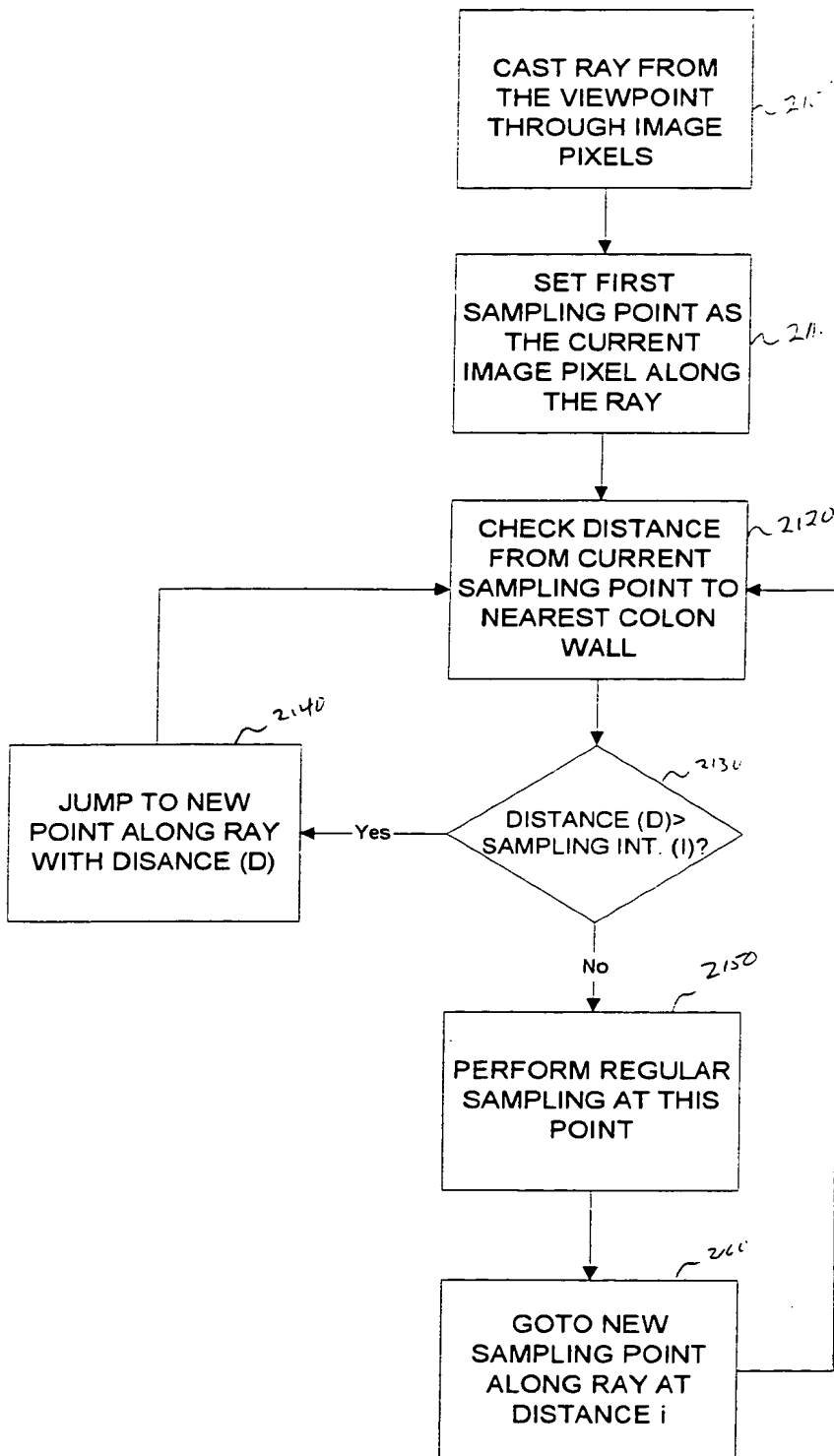
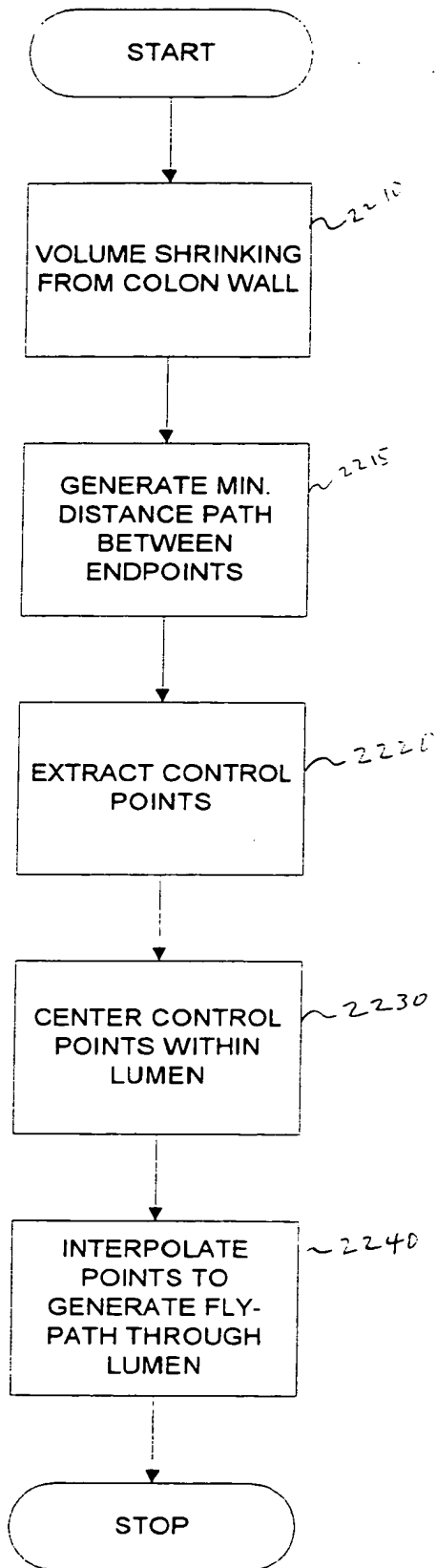
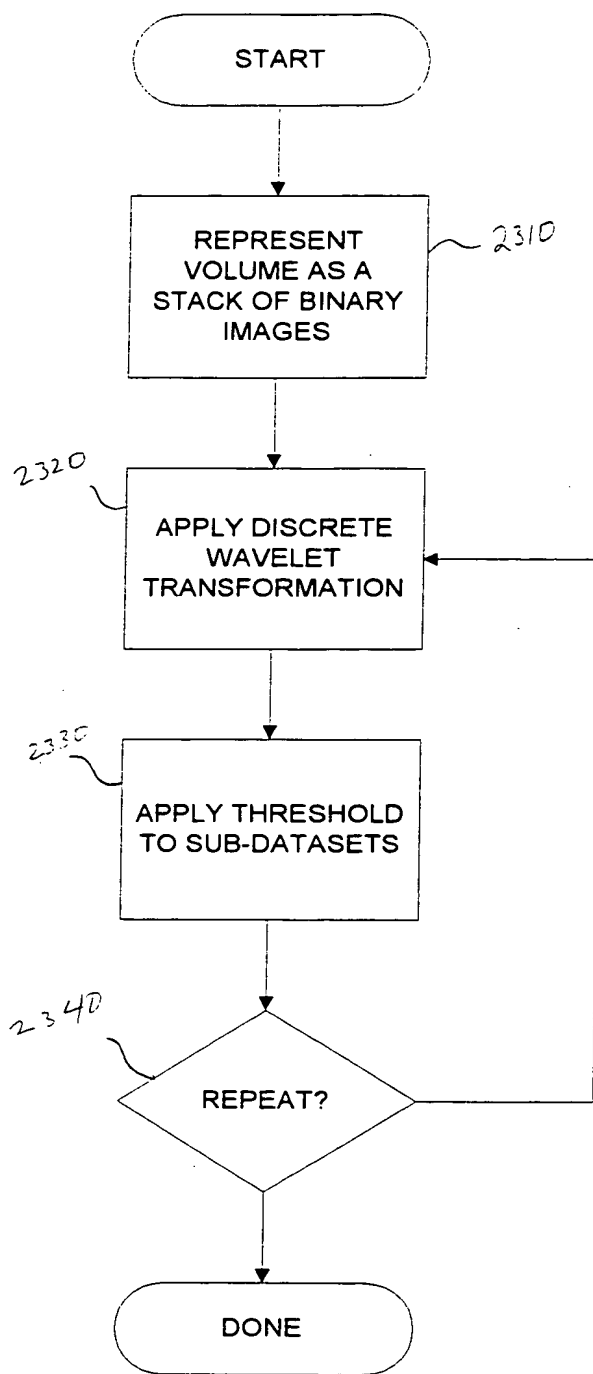


FIG. 22



09/12/2009 09:05:04

FIG. 23



0937460 00001



FIG 24

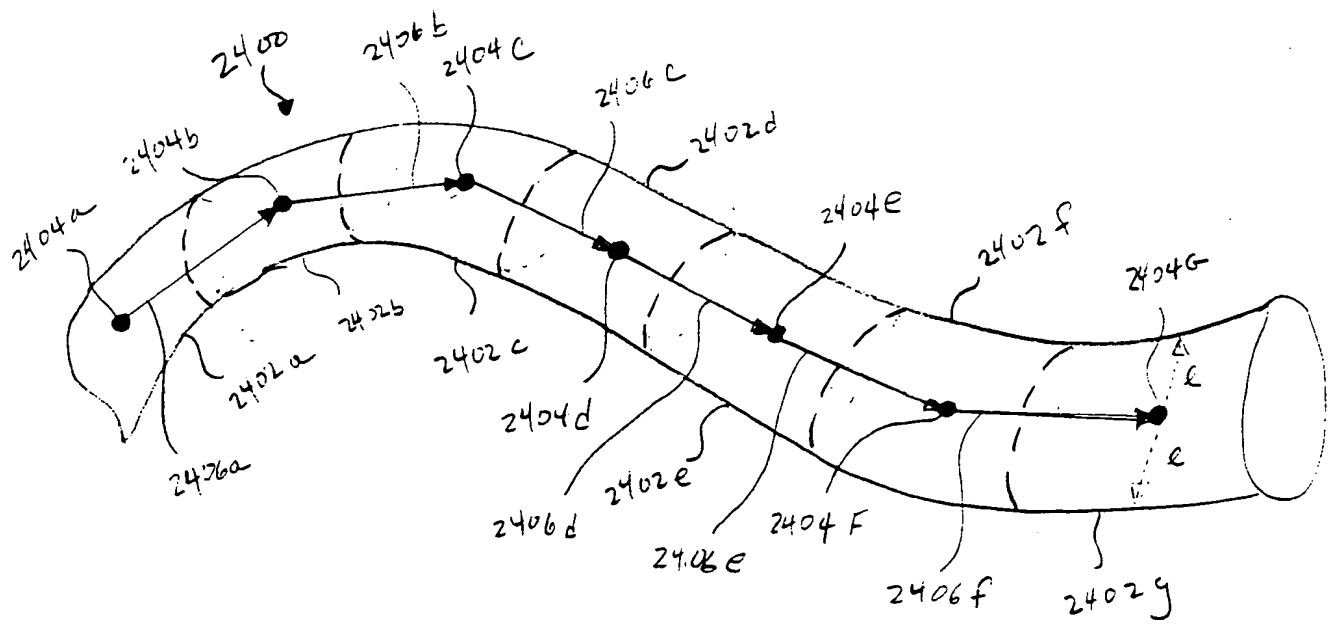


FIG 24

FIG. 25

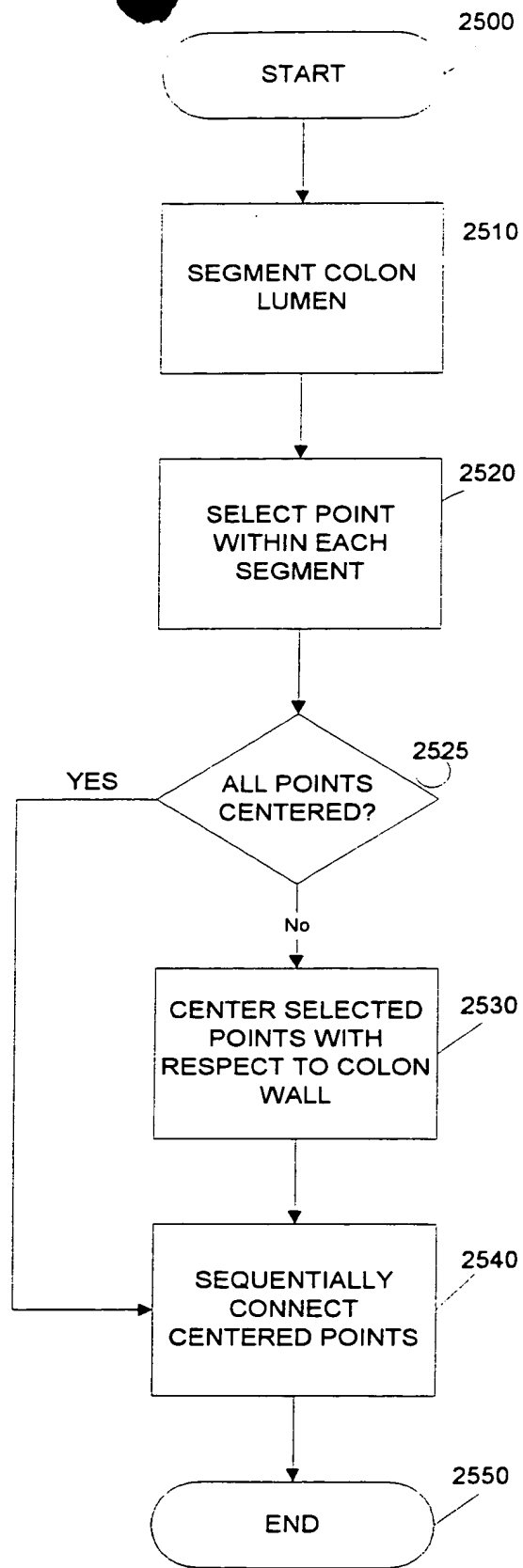


Fig 26

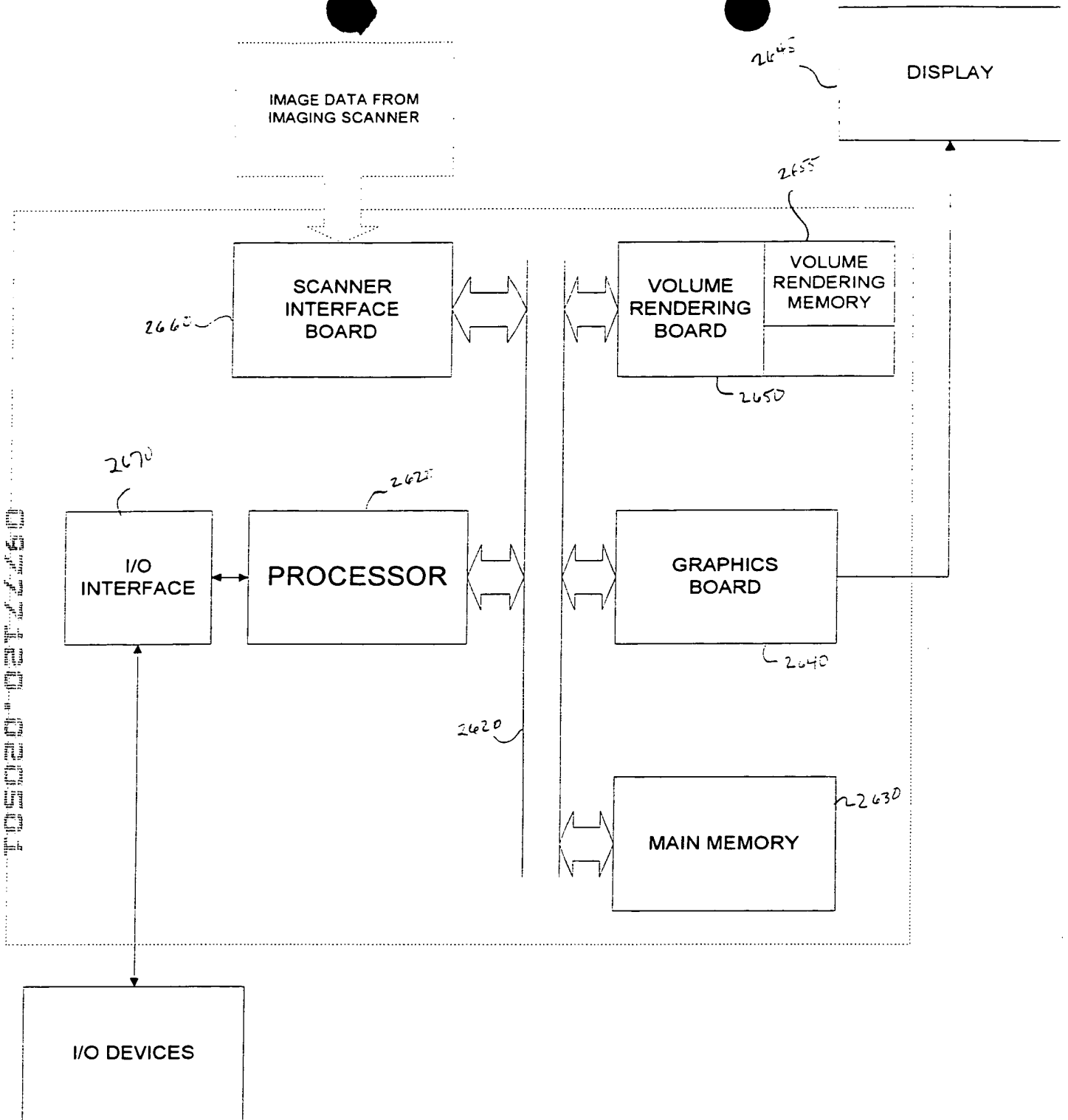


FIG. 27

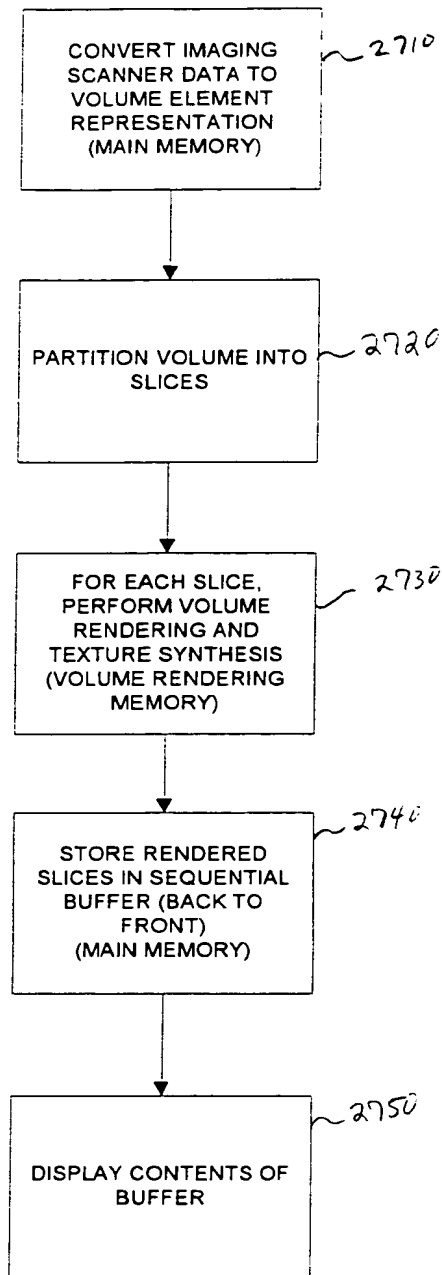


Fig. 28

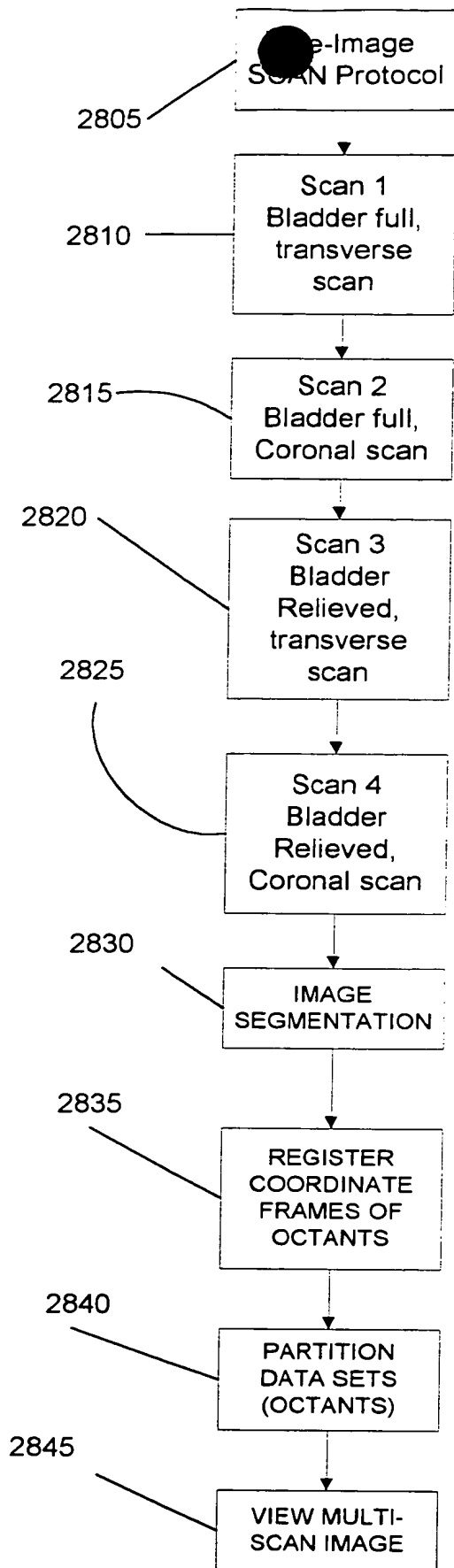


FIG. 29

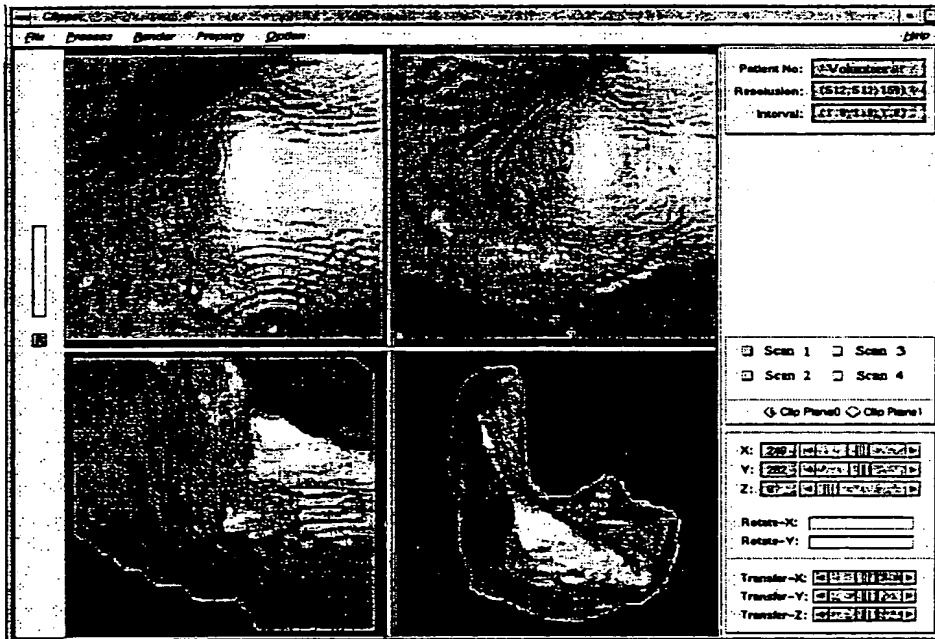
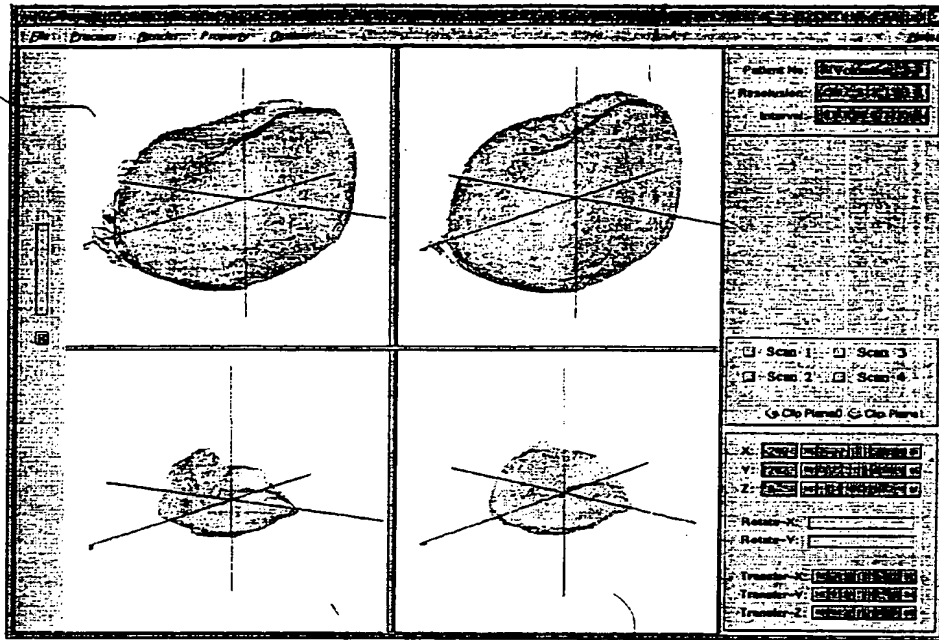


Fig. 32

FIG. 31

2025-03-27 10:00:00

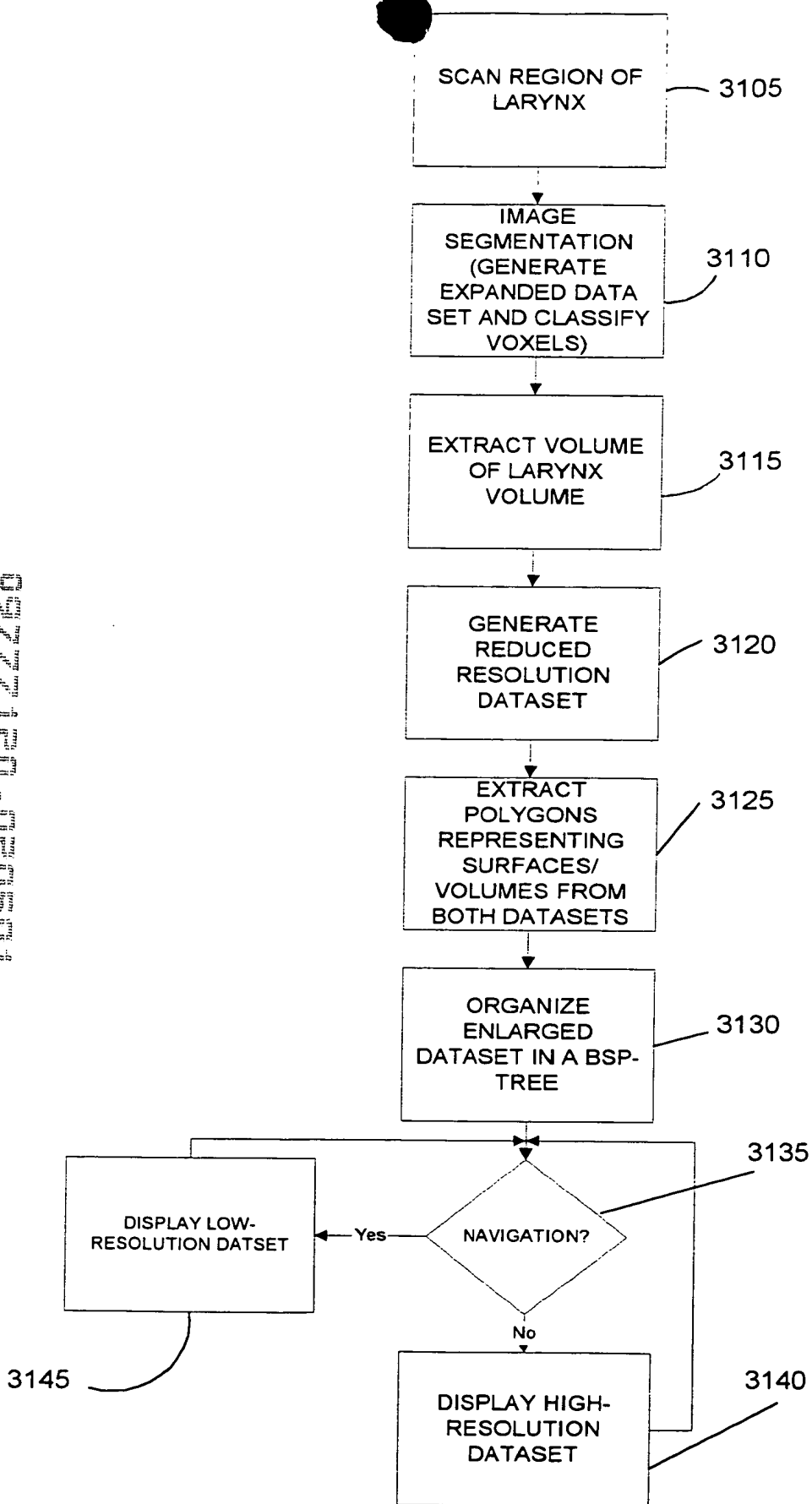


FIG. 32

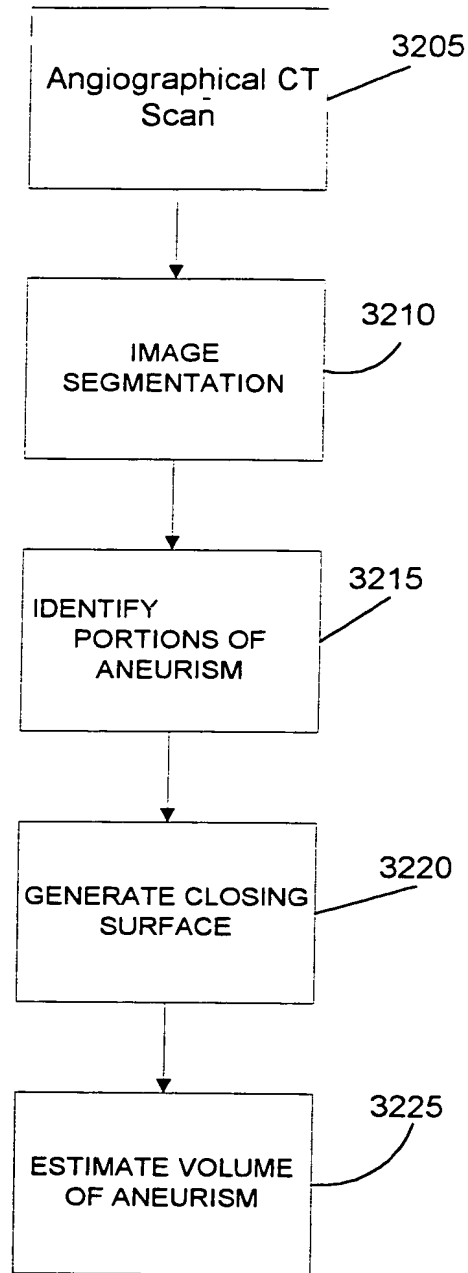




Fig. 306

Fig. 308

Fig. 309

FIG. 306, 308, 309

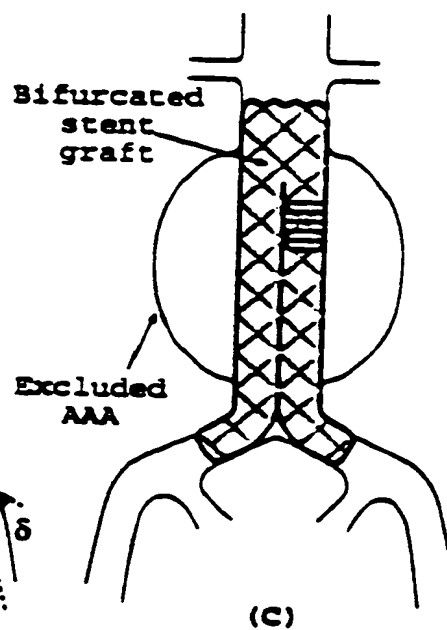
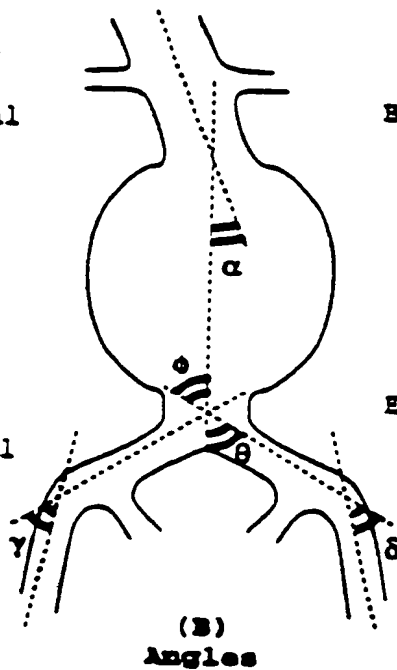
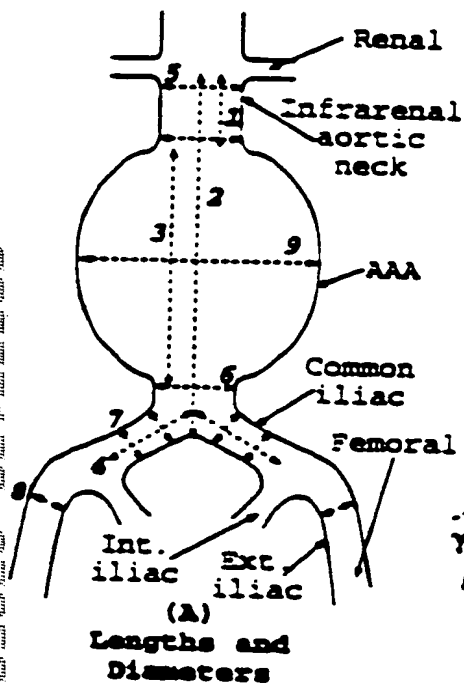


FIG. 3

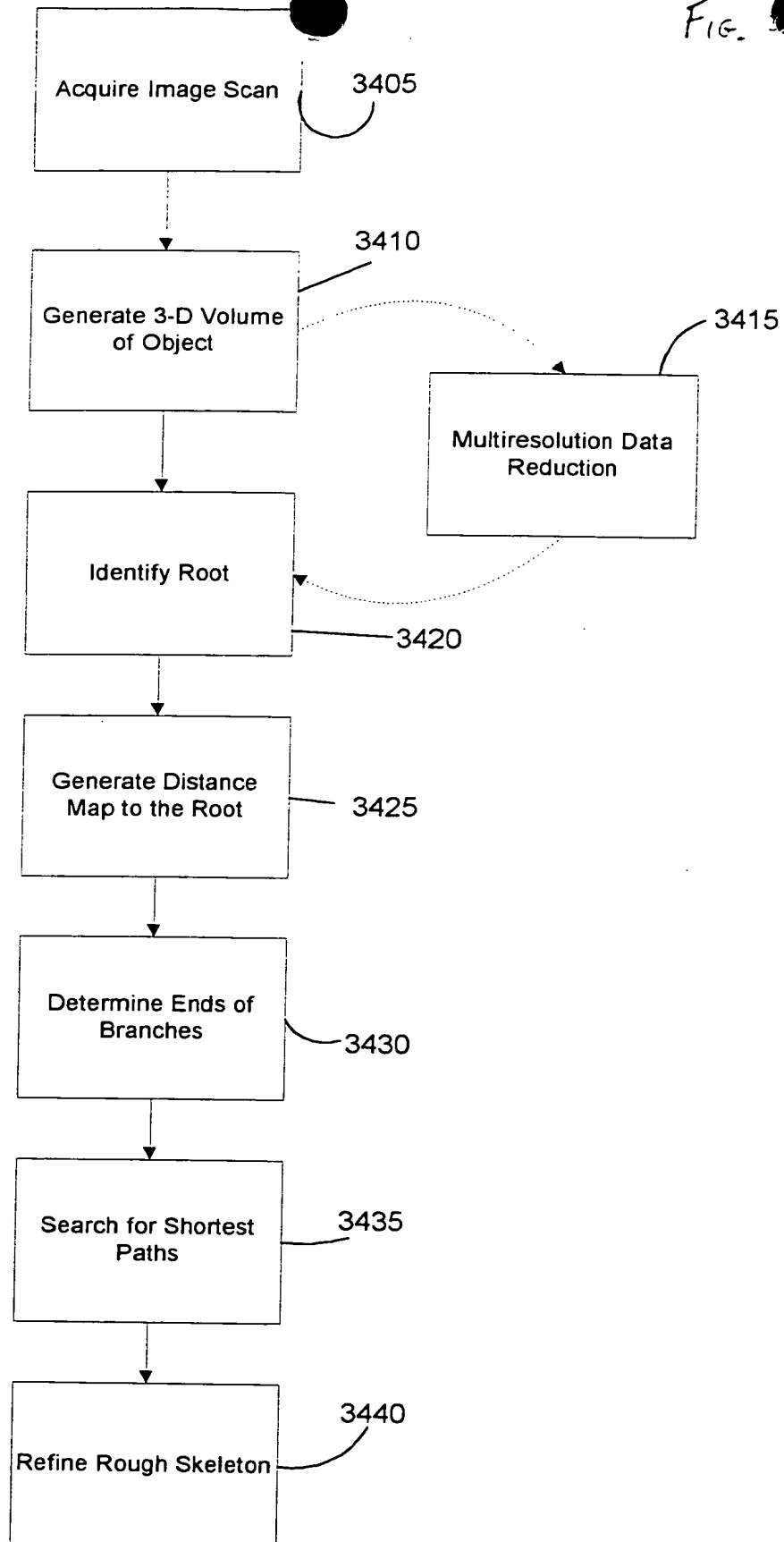


Fig. 35

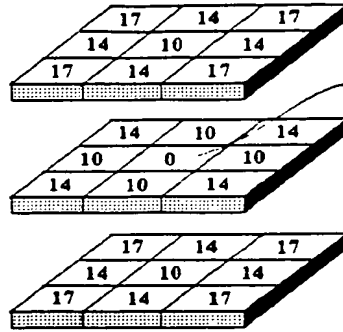


Fig. 36

```

1) Label root voxel with integer 0;
2) Construct a queue and line up the root in the queue;
3) If( There is at least one voxel in the queue )
    Serve the voxel x on the top of the queue:
    For( each of x's 26-connected neighbor voxel y ) {
        If( y in the volume and has not been labeled yet ) {
            Line up the y in the queue;

            /* label the voxel y */
            Set dist = 999999 ;
            For( each of y's 26-connected neighbor voxel z ) {
                If( z in the volume and has been already labeled with an integer of  $n_z$  ) {
                     $d_z \equiv n_z + d(y, z)$ ;
                    where  $d(y, z)$  is 10, 14 or 17 if the Euclidean distance between y and z is
                        1,  $\sqrt{2}$ , or  $\sqrt{3}$ , respectively;
                }
                If( dist >  $d_z$  ) {
                    label y with integer dist ;
                    dist =  $d_z$  ;
                }
            }
        }
    }
    x leaves the queue;
}
Else {
    end of calculating the distance map.
}

```